Naval Research Laboratory

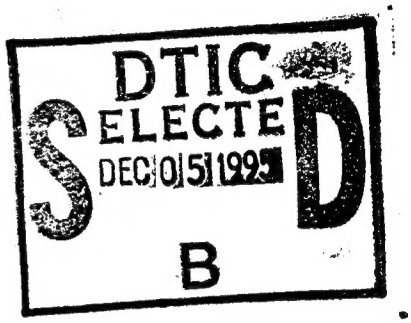
Washington, DC 20375-5320

NRL/FR/5540--95-9795

Security for the Internet Protocol

RANDALL J. ATKINSON

Center for High Assurance Computer Systems
Information Technology Division



November 30, 1995

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 30, 1995	Final

4. TITLE AND SUBTITLE

Security for the Internet Protocol

5. FUNDING NUMBERS

PE - 33140N

6. AUTHOR(S)

Randall J. Atkinson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/FR/5540--95-9795

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Information Security Program Office PD71E
Space and Naval Warfare Systems Command
Crystal City, VA

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distributon unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Lack of widely available security is hindering the growth of the Internet, particularly for commerical users. Two security mechanisms have been designed for use with IPv4 and IPv6. They are an integral component of the IPv6 design but can also optionally work with IPv4. The first mechanism provides source host authentication and information integrity protection without confidentiality and should be exportable and widely deployable. The second mechanism protects the confidentiality of packet contents through the use of encryption. Both mechanisms are designed to be independent of any particular cryptographic algorithm so that new algorithms can be supported in the future without any change to the basic protocols.

14. SUBJECT TERMS

Internet protocol Authentication
Security
Key management

15. NUMBER OF PAGES

14

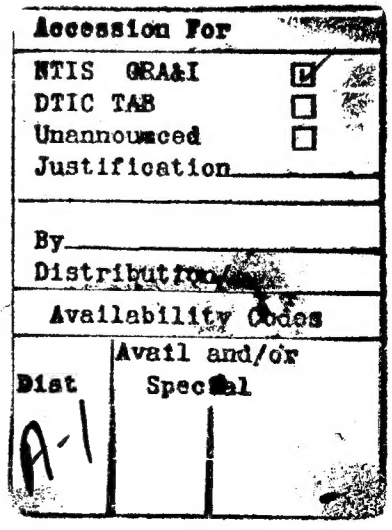
16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

CONTENTS



SECURITY FOR THE INTERNET PROTOCOL

1. INTRODUCTION

During the past decade, the worldwide Internet has grown at nearly exponential rates, not only in North America but also in Europe and Asia [1]. Lack of widely deployed security is a major obstacle to continued growth of the Internet. IPv4 does not yet have a standard set of security mechanisms.[1] This shortcoming of IPv4 is being addressed in both IPv4 and IPv6 with two mechanisms that provide cryptographic security services. The first mechanism, known as the IP Authentication Header, provides cryptographic authentication and integrity without confidentiality. The second mechanism, known as the IP Encapsulating Security Payload, provides confidentiality and possibly integrity through the use of encryption.

In this report, several security related terms are used. Authentication is used to mean that the receiver of a packet can verify the sending system. Integrity is used to mean that the receiver can detect improper modification of the received packet contents. Confidentiality is used to mean that the packet contents are hidden from unauthorized readers.

2. THREATS AND VULNERABILITIES

Most network security work is motivated by concerns about an intruder or attacker gaining unauthorized access to network control information or a computing system on the network. In its early days, Internet users were primarily in the research and education community, so social pressure was used to discourage users from attempting to gain unauthorized access. As the Internet has grown, the user community has become much more diverse, including not only more total users but also many different communities of users. In several parts of the globe, small firms offering Internet access at relatively low cost are appearing.

The first widely publicized Internet break-in was the now infamous "Internet Worm" that exploited a then little-known vulnerability in the **sendmail(8)** mail transfer agent [2]. This vulnerability enabled a remote attacker to gain privileged or "root" access on the system running the sendmail daemon. Over the past several years, a number of additional Internet security flaws have been discovered, primarily implementation errors in network application software such as File Transfer Protocol servers or Hyper-text Transfer Protocol (HTTP) servers [3,4].

However, a more serious problem is that some of the protocols used in the Internet have inherent vulnerabilities [5]. For example, systems using the Network File System (NFS) [6] and Remote Procedure Call (RPC) [7] protocols often make access control decisions based on unauthenticated source IP

---

[1]The "IP Security Option (IPSO)" defined in RFC-1108 provides a sensitivity label field to an IP datagram rather than providing confidentiality, integrity protection, or authentication.

Manuscript approved June 23, 1995.

addresses, which are easily spoofed. Recently, many sites have begun using TCP Wrappers to provide access control lists to network services [8]. This mechanism permits a UNIX computer system to limit which remote systems are permitted to attempt connections to the system running the wrapper and thereby reduces the risk of penetration via network services. Unfortunately, the TCP Wrapper must rely on unauthenticated source addresses as the basis for its access control decisions. Many of these protocol vulnerabilities can be eliminated or significantly mitigated by the addition of cryptographic source authentication at the Internet layer.

While the development of Privacy-Enhanced Mail (PEM) has added confidentiality to personal e-mail, no standard encryption mechanism exists below the application layer in the current Internet. This can make it difficult to use the Internet for electronic commerce.

3. PROPOSED SOLUTION

One method to significantly reduce the risks associated with connecting to and using the Internet is to provide strong cryptologic security as part of the services offered in the Internet. With strong authentication and associated access controls, what break-ins do occur can be more easily tracked. With confidentiality through encryption, many of the passive and active attacks can be entirely precluded. Authentication and integrity protection are provided by the proposed IP Authentication Header. Confidentiality is provided by the proposed IP Encapsulating Security Payload. The balance of this report describes the essential concepts and details of these proposed security mechanisms for use with IPv4 and IPv6.

4. SECURITY ASSOCIATIONS

In order to use cryptography with the Internet Protocol, all of the legitimate parties need to understand which keys, algorithms, and other security-related parameters relate to each packet. This set of security parameters for each communication is known as the *Security Association*. Both of the IP security mechanisms use the same notion of a Security Association. In IP, the Security Association typically includes the key, key lifetime, a Security Parameters Index (SPI), algorithm being used, algorithm mode being used, and whether authentication, encryption, or both are being used. The SPI is an opaque indentifier that is used in conjunction with the Destination Address of the IP packet to determine the particular Security Association in use for the packet. Security Associations are unidirectional, so a typical bidirectional TCP session would have a separate Security Association for each direction. Security Associations are receiver-oriented, meaning that the SPI is assigned by the destination. The SPI is always interpreted in the context of the destination address. This receiver orientation is important to ensure that IP security will be fully compatible with multicast IP, which is also receiver-oriented [9,10]. For systems being used to provide multilevel security, each Security Association will also include the security level, such as *Top Secret*, of that association and the security level of the key for that association. In this case, each security level will have its own set of associations so that there is no chance that *Top Secret* data is accidentally protected with a Secret key instead of the proper *Top Secret* key.

The Security Association is somewhat similar to the Internet's concept of a *flow*. Like flows, a Security Association is bound to a distinct stream of data flowing between a set of systems. However, a single *flow* might have several different Security Associations during its lifetime as key updates occur. The concept of *flows* is described in more detail in Ref. 11.

If all users on the host computer share the same session key for all traffic to some given destination system, then one has host-to-host keying. If, however, each session[2] on the host computer has its own Security Association (and hence session key) for traffic to some given destination system, then one has user-oriented keying. IP security requires that an implementation support user-oriented keying because this precludes a chosen plaintext attack by one user against another user who is using the same computer systems [12]. Also, Jeff Schiller has noted that authentication of principals using applications on end systems requires that processes running applications be able to request and use their own Security Associations [13].

If a user wishes to use a security service, then either an applicable Security Association must already exist (e.g., by manual configuration done in advance) or such an association must be dynamically created (e.g., by a key management protocol). Otherwise, the user will not be able to use the desired security service. Even though a computer system might implement the standard IPv6 security mechanisms, the mechanisms cannot be used if the Security Association does not exist or can not be created when needed. Although this can be a problem for systems that do not implement a key management protocol and do not have preconfigured Security Associations, it does have the nice property that users not desiring to use security will not be forced to use the security mechanisms. Key management is discussed in more detail in Section 7.

5. AUTHENTICATION HEADER

The IPv6 Authentication Header provides IPv6 with exportable cryptographic authentication without confidentiality [14]. The exportability of this mechanism is important because it helps ensure that cryptographic security will be universally available to Internet users. Many potential threats can be mitigated or eliminated by adding cryptographic authentication at the IP layer.

The Authentication Header provides its security features by using a keyed cryptographic hash function that is calculated across the entire packet. The transmitting system calculates the cryptographic checksum and appends it to the outgoing packet as part of the Authentication Header. The receiving system calculates its own version of the cryptographic hash function and compares the result to the value transmitted in the received Authentication Header. If the two match, then the receiving machine can believe that the integrity of the received packet has not been compromised. This allows the receiving machine to trust the contents and header of the received packet, in particular that the source address is genuine and the packet data reliable. The granularity of the authentication provided depends on the granularity of the keying in use. If host-to-host keying is in use, then one can only know that some user on the remote host sent the packet. If user-oriented keying is in use, then one can know that a process connected to the relevant socket on the remote machine sent the packet. The Authentication Header allows for different cryptographic algorithms and authentication data lengths. The rest of this Section describes the structure of the Authentication Header, how it is processed, how the results are used, and some open issues. Figure 1 shows the Authentication Header in relation to the base IP header, the TCP Header, and the TCP data as an example.

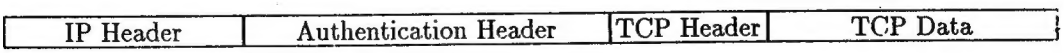


Fig. 1 — Relationship of Authentication Header to IP datagram

[2]The term *session* is used instead of *connection* because not all upper-layer protocols are connection-oriented. A *session* might use any upper-layer protocol.

Authentication Header Format

Figure 2 shows the format of the Authentication Header. In the figure, a single line is 32-bits wide in order to make the format more readable and the alignment more obvious.

Next Header	Payload Length	RESERVED
Security Parameters Index		
Authentication Data		
Authentication Data (cont'd)		

Fig. 2 — Format of IP Authentication Header

The header fields are aligned in a manner friendly to modern computer processors. This alignment was selected to improve protocol processing performance on commonly used 32-bit and 64-bit architectures. The first field in the header is the 8-bit *Next Header* field, which contains the protocol number for the header following the Authentication Header. The next field in the header is the 8-bit *Payload Length* field, which contains the length in 64-bit double words of the Authentication Header being processed. The minimum value for this field is zero double words, which is the degenerate case where no authentication is being used. More typically, this header has a value of two when the keyed MD5 cryptographic hash function is the authentication algorithm used [15]. The third field is a 16-bit *Reserved* field, which is present primarily to preserve alignment. In order to be able to use this for other purposes in the future, this field must be set to all zeros by the sender and ignored by the receiver. The next field, which is the 32-bit *SPI*, is set by the sender to indicate which Security Association is in effect for this datagram. The receiver uses at least this value and the Destination Address of the packet to locate the correct Security Association and hence the information needed to process the incoming packet correctly. The last field contains the *Authentication Data* resulting from the computing cryptographic hash function. This field is variable-length in increments of 64 bits in order to maintain algorithm-independence. The previously described *Payload Length* field dynamically specifies the length used for each packet.

Algorithm independence is important because it means that different user communities can make different security and speed trade-offs by using different authentication algorithms. It also means that should a flaw or vulnerability be discovered in one authentication algorithm, another algorithm can be substituted without changing other aspects of the implementation. For example, if a user community wished to use the NIST Secure Hash Algorithm (SHA) [16] instead of MD5, then the *Authentication Data* field would be three 64-bit double-words (i.e., 192 bits) in length rather than two 64-bit double-words (i.e., 128 bits) in length when MD5 is used. Because the output of SHA is less than 192 bits long, padding bits are appended to the end of the SHA output to make the *Authentication Data* field be an integral number of 64-bit double-words. The values for these padding bits are selected arbitrarily by the sender and are ignored by the receiver. Because the SPI value will indicate which algorithm is in use, the receiver always knows the length of the actual authentication data within the field. All other fields in the Authentication Header are independent of the authentication algorithm in use and remain fixed in length and semantics.

Protocol Processing

When the sender processes an outgoing IP datagram, it examines the sending session state and the system-wide default security configuration to determine whether the Authentication Header is being used for this datagram. Should the session's security configuration and the system-wide default security configuration differ, the more secure value is used by the IP processing engine. In the usual case, the sending IP engine consults the sending user id and destination address to select the appropriate Security Association (and hence SPI value) to use for this datagram. Operating systems seeking to provide multilevel security will also consider the security classification level of the sending process and other mandatory access controls when selecting an outgoing Security Association.

If no applicable Security Association exists and the system supports a key management protocol, then the system will invoke key management to create an association. If no Security Association exists and the system does not support any key management protocol that could be used to create one, the Authentication Header cannot be used. In this last case, the inability to use security should be reported to the application. It is implementation-defined whether the packet is transmitted without the Authentication Header or dropped. Systems seeking to provide multilevel security will drop such packets and in all events will also provide mandatory access controls.

If the Authentication Header is being used and a Security Association exists, the following protocol processing is then performed. The sending system calculates authentication data for an outgoing IPv6 datagram by applying a cryptographic hash function over a secret key and the parts of the IPv6 packet that do not vary in transit. The details of how the key is used in calculating the authentication data depend on the cryptographic transform being used.

When the default Keyed MD5 cryptographic hash function is being used, the secret key is both prepended and appended to the packet data being authenticated for the purpose of calculating the *Authentication Data*. The resulting authentication data is then transmitted as the *Authentication Data* field in the Authentication Header of the IP datagram, and the other fields of that header are appropriately filled in.

The receiver of a packet containing an Authentication Header uses the Destination Address of the packet and the SPI value to locate the relevant Security Association information. It then recalculates the authentication data using the algorithm and key for that Security Association. This locally calculated authentication data is compared with the transmitted authentication data. If the two match, the packet is considered authentic, and if they do not match, then the packet is not considered authentic and appropriate security error handling occurs (e.g., auditing). The Destination Address is used for the demultiplexing rather than the Source Address because IP multicasting is receiver-oriented and has the multicast group as the Destination Address [10]. If the Source Address were used for demultiplexing, then the security mechanisms would not work well with multicast IP traffic.

Open Issues

Widespread application of the Authentication Header is likely to be impacted by the performance impacts of published cryptographic hash functions. Experimental software-based implementations of MD5 reportedly can process data at speeds of 50 to 120 Mbps on fast commercial 64-bit RISC processors. The performance limit of hardware-implementations of MD5 is not yet clear. Additional study of specific cryptographic hash functions and their performance impacts and performance limits is needed.

6. ENCAPSULATING SECURITY PAYLOAD

The IP Encapsulating Security Payload (ESP) provides optional confidentiality and integrity to IP datagrams through the use of encryption [17]. Although a default encryption algorithm and transform have been specified to ensure interoperability throughout the Internet, the mechanism is designed to be algorithm independent. While similar in some respects to the U.S. Government's SP3D security protocol, ESP is different in other respects. For example, SP3D includes a security label as part of the encryption mechanism, while ESP separates that out and uses implied security labels rather than having explicit security labels. Another difference is that ESP has a minimal header, which makes parsing and protocol processing simpler. Figure 3 shows the relationship of ESP to an overall IP datagram, omitting optional headers for clarity.

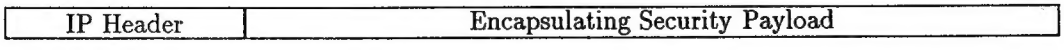


Fig. 3 — Relationship of ESP to IP datagram

ESP Header Format

This section describes the syntax and semantics of the ESP header. Figure 4 shows the format of the IP ESP.

Fig. 4 — Format of the IP ESP

The ESP has a single cleartext field. This is the 32-bit *SPI* field, which was described earlier. The syntax and semantics of an SPI are the same for both this encryption mechanism and also the authentication mechanism described above. The remainder of the ESP bit format depends on the cryptographic transform that is in use. This provides complete algorithm independence, which is an important design feature.

There are several additional fields that are present when the default DES CBC transform is in use [18]. First, there is a 64-bit *Initialization Vector* field that comes immediately after the SPI field. After that there is the protected data that has been decrypted. After decryption, the last octet of the decrypted data contains a *Next Header* field that indicates which protocol is present at the start of the decrypted protected data. The next to the last octet after decryption contains the *Pad Length* field, which contains the number of octets of padding data at the end of the protected data, not counting the *Pad Length* or *Next Header* fields. The protected data can be an entire IPv4 or IPv6 datagram, an IPv6 optional header, or it can be any upper-layer protocol (e.g., TCP, UDP, ICMP).

After decryption, when the default DES CBC transform is in use and the protected data contained a TCP header and data, the *Protected Data* for the above packet has the following layout:

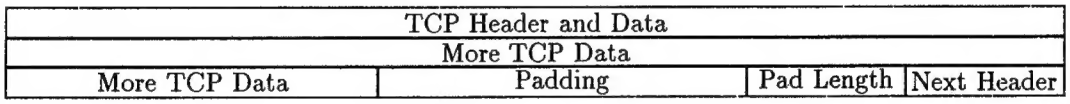


Fig. 5 — Detailed example of the ESP *Protected Data* field

The presence of TCP in Fig. 5 is just for illustration. Any set of IPv6 headers and user data or an entire encapsulated IP datagram might be contained within the *Protected Data* part of the ESP. Padding is necessary when the protected data (including the *Pad Length* and *Next Header* fields) are not a full cryptographic block size because DES CBC only encrypts full blocks. For DES, the cryptographic block size is 64 bits [19].

Protocol Processing

Protocol processing for IP packets containing an ESP is very similar to that for IP packets containing an Authentication Header.

When the sender processes an outgoing datagram, it examines the sending session state and the system-wide default security configuration to determine whether the ESP is being used for this datagram. Should the session's security configuration differ from the system-wide default security configuration, the more secure configuration is used. In the usual case, the sending IP engine then consults the sending socket information and the destination address in order to select an appropriate Security Association (and hence SPI value) to use for this datagram. Systems providing multilevel security will also consider the security classification level of the sending process and other mandatory access controls when selecting an appropriate Security Association.

If no applicable Security Association exists and the system supports a key management protocol, then the system will invoke key management to create an association. If no Security Association exists and the system does not support any key management protocol that could be used to create one, then ESP cannot be used. In this last case, the inability to use security should be reported to the application. It is implementation-defined whether the packet is transmitted without the ESP or dropped in this case. Systems seeking to provide multilevel security will drop such packets and will always apply additional mandatory access controls.

If the ESP is being used and a Security Association exists, the sending system then encrypts the appropriate portion of the outgoing datagram using the algorithm and key belonging to that Security Association. The details of the encryption process depend on the particular algorithm and algorithm mode being used.

The receiver uses the Destination Address and SPI value to locate the appropriate Security Association information. It then performs the decryption using the key and algorithm for the located Security Association. If decryption fails, then the results will probably not be parsable and so the failure can usually be detected. Failed decryption attempts are logged and the failed packet information is then discarded. After successful decryption, the padding is removed from the protected data and then the protected data is processed as if it had not been encrypted. The value of the *Next Header* field is used to determine which protocol or header begins at the start of the decrypted protected data. The system's normal protocol processing routines can be used to process this protected-data from this point.

Use of ESP with the Authentication Header

The usual combination of these two security mechanisms places the Authentication Header in the cleartext portion of the IP datagram in order to permit authentication of all of the cleartext fields. This placement also provides strong integrity checking for the data encrypted by ESP. If the authentication check should fail at the receiver in this case, then the ESP decryption need not be attempted. This combination is illustrated next. This combination might also be useful if ESP should be used with a cryptographic transform that provides confidentiality without authentication or integrity.

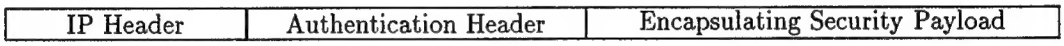


Fig. 6 — IP datagram containing both ESP and the IP Authentication Header

Open Issues

The precise performance impacts of encryption are not clear. It appears likely that many IPv6 implementations will use software encryption. This might significantly reduce the network performance seen by users because any kind of encryption will introduce significant latency. Hardware-based encryption will be necessary for higher bandwidth links and for users requiring low latency. Fortunately, it is feasible to build very high speed DES chips, so one can have encryption while retaining good network performance [20].

Some have suggested that Triple DES be used instead of single DES in Cipher-Block Chaining (CBC) mode in order to increase the effective key size. Triple DES may become common in the future, but Triple DES is slower than normal DES CBC, making DES CBC a better choice for the default encryption algorithm. It is certain that in the future different user communities will want to use other algorithms with ESP. The algorithm-independence of the ESP design should make this straightforward.

On some systems, it might be difficult to implement support for separate Security Associations for each session between a set of communicating systems. NRL is implementing such support inside 4.4 BSD, but operating system internal issues might arise with other systems. Single-user systems such as DOS will not gain much benefit from having separate Security Associations for each concurrent session.

7. KEY MANAGEMENT

The IP security specifications require that manual key distribution be supported by all implementations. This is important because not all users will have or desire to use a key management protocol. Manual key distribution can work well within a LAN, or within a modest sized set of communicating systems (e.g., routers within a single administrative domain or researchers at different sites that are collaborating), and permits some use of the above security mechanisms even before a scalable key management protocol is standardized.

A scalable key management protocol will be needed for these mechanisms to be widely successful in providing cryptographic security to IPv6 users. Such a key management protocol based on asymmetric cryptography is currently being developed within the Security Area of the Internet Engineering Task Force (IETF) [21]. The general approach is to use a hybrid Diffie-Hellman algorithm [22] similar to van Oorshot's Station-to-Station algorithm [23]. This algorithm would obtain signed public keys associated with hosts from the Internet's Domain Name System (DNS) [24]. The public keys from the DNS would be signed with the well-known public key of a key certification authority. The parties to the Diffie-Hellman exchange would then use these signed public keys to secure the exchange of secret session keys. The key management protocol would also negotiate the other security parameters in use, for example, the cryptographic algorithm, Security Parameters Index, and security classification level (if any). This use of signed public keys from the DNS to bootstrap into the Diffie-Hellman exchange eliminates the well-known "man in the middle" vulnerability of Diffie-Hellman. This approach can also provide perfect forward secrecy to the session keys, which is especially important since most of the computers on the Internet do not currently use high-assurance operating systems.

Scalable multicast key distribution remains a research area. Some work on this topic is underway within the Internet community. One approach uses centralized key distribution centers [25]. Another approach takes advantage of routing infrastructure to distribute keys [26]. It is not yet clear whether either of these approaches will be standardized or widely deployed.

8. APPLICATION USE OF SECURITY

The Authentication Header provides data origin authentication and integrity protection for IPv6 datagrams. Sites currently using address-oriented access controls to network services [8] can use the Authentication Header to provide authentication to the data used for those access control decisions. Also, this mechanism can be used to detect unauthorized modification of the packet and its contents by a network intruder. In the case of networks handling data with sensitivity labels,[3] the Authentication Header can be used to cryptographically bind the labeling information to the packet. This can reduce the risk of the packet's labeling information being modified while the packet is in transit. This is a significant improvement over the current practice of using security labels even though IPv4 packets have lacked a cryptographic binding between the packet and its label [27].

ESP provides confidentiality and, depending on the ESP cryptographic transform in use, integrity to data in IP packets. It can eliminate a broad class of attacks on the network and network users. Passive attack on cleartext passwords are precluded [28]. Active attacks on connections are also precluded [29]. Traffic analysis, although not precluded, is made more difficult by the provided encryption. Availability of confidentiality facilitates increased commercial and consumer use of the Internet. For example, credit card numbers sent in encrypted packets would be protected against passive attack and consequent credit fraud.

Applications will need to have standard methods to request security services and to determine which security services are actually being provided. Work is underway to develop a security application programming interface that is compatible with the widely used Berkeley Sockets networking interface [30]. Once such an API is available, then applications can selectively use the security services that they need. Also, the applications

[3]For example, "Finance Department Only"

could let the user select which services are appropriate on a session by session basis. For security to be widely useful, it must be widely available to applications, and users must have appropriate control over when security is used and when it is not used.

9. RESIDUAL RISKS

Significant risks remain in any operational network, even when cryptographic security services are designed into the network. Implementation vulnerabilities cannot be completely eliminated by using IP-layer encryption or authentication. If the authentication or encryption algorithm in use is compromised, then it can not provide the intended protection and would need to be replaced by a stronger algorithm. If the cryptographic keys were known to an attacker, either from a flaw in the key management protocol or by insider attack, then again the intended security services will not be provided. There is some cause for concern that any key management protocol might have a subtle flaw discovered long after it is widely deployed [31,32,33]. Also, over time it is likely that new cryptologic algorithms will need to replace those currently proposed, due either to newly discovered cryptanalysis techniques [12] or to increased brute-force risks [34]. Additionally, some trust in an operating system's ability to correctly associate a user or a user's application with the correct security association is required. Strong separation between multiple users on a single machine will depend on this correctness as well as other operating system features. In some cases, electronic mail for example, the needed security services can only be provided within the application layer. While IP security can significantly reduce security risks, it can not solve or address all of the security issues in networks or distributed systems.

10. SUMMARY

This report has described two security mechanisms being designed for use with both IPv4 and IPv6. Use of these mechanisms can significantly reduce network-related risks. Because these mechanisms are algorithm-independent, new and improved cryptographic algorithms can be incrementally deployed in the future without the need to redesign the security mechanisms.

11. ACKNOWLEDGMENTS

This work was sponsored by Mike Harrison and Tim McChesney of the Information Security Program Office (PD71E) of the U.S. Space and Naval Warfare Systems Command. Paul Crepeau, Richard Hale, and Joe Macker of NRL provided detailed comments on an earlier version of this report.

12. REFERENCES

1. M. Lottor, "Internet Growth (1981-1991)," Jan. 1992, RFC-1296.

2. E.H. Spafford, "The Internet Worm: Crisis and Aftermath," *Commun. of the ACM*, **32**(6) 678-698 (June 1989).

3. C. Stoll, *The Cuckoo's Egg* (Simon and Schuster, New York, NY, 1990).

4. "ftpd Vulnerabilities," Computer Emergency Response Team, Apr. 1994, CA-94:08.

5. S.M. Bellovin, "Security Problems in the TCP/IP Protocol Suite," *ACM Computer Commun. Rev.* **19**(2) (Mar. 1989).

6. Sun Microsystems Inc., "NFS: Network File System Protocol Specification," Mar. 1989, RFC 1094.

7. Sun Microsystems Inc., "RPC: Remote Procedure Call Specification," Version 2, June 1988, RFC-1057.

8. W. Venema, "TCP WRAPPER: Network Monitoring, Access Control and Booby Traps," Proc. of the Third Usenix UNIX Security Symposium, USENIX Association, Sept. 1992.

9. S. Deering and B. Hinden, "IPv6 Specification, Feb. 1995" (Internet Draft).

10. S. Deering, "Host Extensions for IP Multicasting," Aug. 1989, RFC-1112.

11. D. Clark, S. Shenker, and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism," Proc. ACM SIGCOMM '92, Association for Computing Machinery, Aug. 1992.

12. E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard* (Springer-Verlag, New York, NY, 1993).

13. J.I. Schiller, Re: IPv6 Security, Mar. 1995 (Electronic mail message).

14. R. Atkinson, "IP Authentication Header," May 1995 (Work in Progress).

15. R.L. Rivest, "The MD5 Message-Digest Algorithm," Apr. 1992, RFC-1321.

16. U.S. NIST, "Secure Hash Standard," July 1995, FIPS PUB 180.

17. R. Atkinson, "IP Encapsulating Security Payload," May 1995 (Work in Progress).

18. P. Metzger, P. Karn, and W. Simpson, "The ESP DES-CBC Transform," Apr. 1995 (Work in Progress).

19. U.S. NIST, "Data Encryption Standard," Jan. 1977, FIPS PUB 46.

20. H. Eberle, "A High-Speed DES Implementation for Network Applications," In *Advance in Cryptology — CRYPTO '92 Proc.*, Springer-Verlag, New York, NY, 1993, pp. 527-545.

21. P. Karn and W. Simpson, "The Photuris Session Key Management Protocol," Mar. 1995 (Work in Progress).

22. W. Diffe and M. Hellman, "New Directions in Cryptography," *IEEE Transaction on Information Theory*, IT-**22**(6), 644-654 (Nov. 1976).

23. W. Diffe, P.C. van Oorschot, and M.J. Wiener, "Authentication and Authenticated Key Exchanges," in *Designs, Codes, and Cryptography*, **2** Kluwer Academic Publishers (1992) pp. 107-125.

24. D.E. Eastlake and C.W. Kaufman "Domain Name System Protocol Security Extension," Jan. 1995 (Work in Progress).

25. H. Harney et al., "Group Key Management Protocol," Oct. 1994 (Internet Draft).

26. A. Ballardie, "Scalable Multicast Key Distribution," Nov. 1994 (Work in Progress).

27. S. Kent, "U.S. DoD Security Options for the Internet Protocol," Nov. 1991, RFC-1108.

28. N. Haller and R. Atkinson, "On Internet Authentication," Oct. 1994, RFC-1704.

29. "IP Spoofing Attacks and Hijacked Terminal Connections," Computer Emergency Response Team, Jan. 1995, CA-95:01.

30. D.L. McDonald, "IP Security API for BSD Sockets," May 1995 (Work in Progress).

31. R.M. Needham and M.D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," *Commun. of the ACM*, **21**(12), 993-999 (Dec. 1978).

32. D.E. Denning and G.M. Sacco, "Timestamps in Key Distribution Protocols," *Commun. of ACM*, **24**(8) 533-536 (Aug. 1981).

33. M. Blaze, "Protocol Failure in the Escrowed Encryption Standard," Proc. of 2nd ACM Conf. on Comp. and Commun. Security, Fairfax, VA (Nov. 1994) Association for Computing Machinery.

34. M.J. Weiner, "Efficient DES Key Search," Tech. Rep. 244, School of Computer Science, Carleton University, May 1994.